Amendments to the Specification

Please **Amend** the paragraph beginning at page 4, line 34 as follows:

It is advantageous to the invention if the states of DRAM memory banks are depicted by associated state machines. This makes it possible to control all memory banks independently of one another. For each access operation, the state machines receive the type of transfer (read or write), the row number and the column number. By observing particular rules for time coordination, they control the memory banks by sending commands to the command scheduler. In this case, each channel is connected to the state machine which controls the associated memory bank. If ~~a channel is able to access~~ a plurality of memory banks can be accessed via a channel, a network is required. The command scheduler ensures that the same memory bank is not addressed a plurality of times in succession. Between two access operations to a memory bank, an access operation to another memory bank is always effected. Alternatively, however, two successive access operations to a memory bank are permitted if they are made to the same row in the memory bank, which means that no waiting times arise as a result of the activation or precharging. The priority allocation sorts the pending commands according to their ability to start a new burst in such a way that optimum use of the DRAM data bus is achieved. This means that read and write commands have a high priority, followed by activation commands, which are a prerequisite for read or write commands. Precharging commands are given the lowest priority, since they are not part of the current transfer. Precharing commands are required only for successive transfers, and they can therefore be delayed. If all bursts have a length of four or more clock cycles, the workload on the command lines is small enough to transmit commands with low priority without a long delay. To stipulate the order of the waiting commands, the commands need to be analysed, grouped and sorted according to their ability to start a data transfer as quickly as possible:

Please **Amend** the paragraph beginning at page 7, line 12 as follows:

Figure 2 shows the block diagram of an inventive memory controller using the example of an SDRAM controller 1 in a system with three channels: ~~a CPU connected via~~ an AMBA bus 8 for connecting a CPU, and also an input 6 and an output 7 for a real-time data stream. Each memory bank 21, 22, 23, 24 in the SDRAM module 2 has an associated state machine 41, 42, 43, 44 in a memory bank control unit 4, which depicts the respective state of the memory bank 21, 22, 23, 24 and is responsible for observing the waiting times and the correct state sequence. These state machines 41, 42, 43, 44 transmit their commands for the memory banks 21, 22, 23, 24 to a command scheduler 3 (command bus scheduler) which watches over the allocation of the external command and data bus. In each clock cycle, the command scheduler 3 transmits a command selected according to priority to the DRAM module 2. The state machines 41, 42, 43, 44 obtain their transfer orders directly [[from]] via the three channels (input 6, AMBA 8 and output 7), which are forwarded by a memory bank scheduling unit 5 on the basis of their address and priority to the appropriate memory bank 21, 22, 23, 24. The memory bank scheduling unit 5 contains a network in order to allow all channels 6, 7, 8 to access all memory banks 21, 22, 23, 24. During a read access operation to the storage medium, ~~the input channel 6 accepts~~ the data from an ECC (Error Correction Code) unit (not shown) are accepted via the input channel 6, and ~~the output channel 7 forwards~~ the data are forwarded via the output channel 7 to an ATAPI block (not shown). Both channels 6, 7 contain FIFOs (not shown) in order to prevent the flow of data from being held up. During a write access operation to the storage medium, ~~the input channel 6 receives~~ the data from the ATAPI block are received via the input channel 6, and ~~the output channel 7 forwards them~~ are forwarded to the ECC unit via the output channel 7. An AMBA (Advanced Microcontroller Bus Architecture) slave, which is integrated in the [[The]] AMBA ~~(Advanced Microcontroller Bus Architecture)~~ channel 8 ~~comprises an AMBA slave~~, [[which]] additionally permits access to a register file 34 as well (see Figure 3). It contains a read cache and a write cache (not shown)

in order to reduce the blocking time for the AMBA bus. Since the internal state of each of the four SDRAM memory banks 21, 22, 23, 24 is depicted by a separate state machine, merely accessing a state machine 41, 42, 43, 44 may result in competition by the three channels 6, 7, 8 which is not handled by an upstream scheduling algorithm. However, it is possible to ensure that this competition situation arises only rarely. For this reason, the real-time data stream of the sector data is granted priority over the ARAM access operations in this case.

Please **Amend** the paragraph beginning at page 9, line 6 as follows:

If the global command sent is an NOP (No Operation), then a memory bank command can be transmitted to the DRAM module 2. Memory bank commands are transmitted on the basis of a static priority allocation for commands and a dynamic priority allocation for channels 6, 7, 8, which are executed by a priority allocation unit 33. In this case, the priority of commands is higher than that of the channels 6, 7, 8. This means that first a command type is chosen and, if there are a plurality of channels 6, 7, 8 over which ~~wish to send~~ this command is sent, then the channel 6, 7, 8 is chosen, over which sending this command is allowed ~~to send the command is chosen~~.

Please **Amend** the paragraph beginning at page 9, line 33 as follows:

The dynamic prioritization ~~prioritisation~~ of the channels 6, 7, 8 is effected by an algorithm as is shown in the form of a final state machine, i.e. a state diagram, in Figure 4. The algorithm shown controls the access operations of a CPU via an AMBA-AHB (Advanced High-performance Bus) and a real-time data stream [[with]] via its two channels (input and output). The states represent the priority levels. In this case, the channel shown in the top state has the highest priority and the bottom channel has the lowest priority. The state transitions represent

the channel via which ~~can ultimately start~~ a read or write burst can be started. A possible additional channel for a flash controller, via which for example ~~for sending~~ firmware etc. is sent, is not included, since it does not compete with the other channels.

Please **Amend** the paragraph beginning at page 10, line 10 as follows:

As can be seen, the Input → AMBA → Output → AMBA sequence is always observed when all three channels are active. If no command can be sent via the channel with the highest priority ~~is not able to send a command~~, since [[it]] the channel is currently not active or ~~its command~~ the command to be sent has too low a priority, the channel via which ~~can ultimately send its~~ a command can be sent is given the lowest priority in the next clock cycle. At the same time, however, it is ensured that the AMBA channel 8 is given the highest priority in the next clock cycle if it does not have the highest priority in the current clock cycle and another channel has an opportunity. Once the AMBA channel 8 has been given the highest priority, it loses it again only when ~~it can send a command~~ a command was sent via the AMBA channel 8. This ensures the lowest possible latency for the ARM. The state diagram shown guarantees short delay times for the CPU access operations, since the AMBA channel 8 is given the highest priority after every burst via the input channel 6 or the output channel 7. In addition, the state diagram ensures fair use of the data bus and alternating access operations to the memory banks 21, 22, 23, 24. The algorithm is designed for CPU access operations with high priority and hence low latency given simultaneous guaranteed data throughput for the real-time data stream. The data throughput is stipulated by the length of the read and write bursts through the input channel 6 and the output channel 7 upon data transfer from and to the DRAM 2.

Please **Amend** the paragraph beginning at page 11, line 1 as follows:

The decision regarding via which channel sending of a command is permitted ~~to send its command~~ will be explained below with reference to the flow chart shown in Figure 5. If a memory bank FSM (Final State Machine) wishes to terminate its burst, this has the highest priority. If, consequently, a Burst Terminate (BST) has been found after the start 9 during the analysis 10, the burst in progress is aborted. This can be done in two ways: first by simply forwarding 12 the Burst Terminate command, secondly by starting 14 a new burst. Before a Burst Terminate is now sent, a check 11 is performed to determine whether a Read or Write command is likewise waiting to be sent. If this is the case, this command is sent 14 instead of the Burst Terminate. During the analysis, there is merely a check to determine whether at least one Read or Write command is present. For this reason, before the Read or Write command is sent 14, the dynamic priority allocation for the channels is used to check via which ~~is the channel which has~~ a Read or Write command is to be sent and has the highest priority. [[This]] The command is then sent via this channel ~~is then permitted to send its command~~. The choice of channel is communicated to the priority allocation, which thus changes to a new state with a new distribution of priorities in the next clock cycle. The only restriction for replacement of the Burst Terminate command with a Read of Write command is that a Read Burst cannot be terminated by a Write Burst, since otherwise the memory controller 1 and the DRAM module 2 are driving the data bus simultaneously. When a lower clock frequency is used, this restriction can be bypassed, however, since the hold time for the outputs of the DRAM is constant and is not dependent on the clock frequency.

Please **Amend** the paragraph beginning at page 11, line 34 as follows:

If no Burst Terminate command is present, then the presence of Read or Write commands is checked 13 and, if they are present, the command is transmitted 14 on the basis of the priority allocation. If no Read or Write commands are present either, then there is a check 15 for Activate commands. If an Activate

command of this type is waiting, it is transmitted 16. If there is no Activate command, there is a check 17 for Precharge commands. Any Precharge command which is present is transmitted 18. Should ~~no channel 6, 7, 8 or~~ no memory bank 21, 22, 23, 24 wish to send a command 19 and no command be sent via a channel 6, 7, 8, an NOP (No Operation) is transmitted. If a command is sent to the DRAM module 2, the memory bank FSM from which this command comes is informed by a signal, as a result of which it changes to a new state in the next clock cycle.